УДК 519.6

М. Ю. Андрамонов

РЕШЕНИЕ ЗАДАЧ КВАЗИЛИНЕЙНОЙ АЛГЕБРЫ*

1. Введение

В данной статье предлагается общая схема метода для решения задач квазилинейной алгебры. Такие задачи возникают в негладком анализе (см. [1, 2]), в частности, при решении задач оптимизации негладким методом Ньютона (см. [1]), в глобальной оптимизации.

Задача квазилинейной алгебры представляет собой одну из задач дизъюнктивного программирования, которое является перспективным направлением в области численных методов оптимизации (см. [3–5]). Задача дизъюнктивного программирования является задачей с линейными ограничениями (имеются обобщения на нелинейный случай), которые связаны логическими условиями. В результате приведения к конъюнктивной нормальной форме условия представляются в виде совокупности дизъюнкций, каждой из которых соответствует допустимый многогранник.

Простейшим способом решения данной задачи является ее сведение к задаче целочисленного программирования. Численный опыт показывает, что подобный подход эффективен только для малой размерности.

Вследствие этого необходимо использовать специфику задачи. Метод состоит в разбиении допустимого множества, последовательной релаксации элементов разбиения, выборе наиболее перспективного из них и отсечении заведомо неоптимальных областей.

2. Постановка задачи

Требуется решить следующую систему уравнений:

$$\begin{array}{ll} x \in X_1 = \{x : \max\limits_{i \in Q_1}[a_{i1}, x] = b_1\}, & \\ x \in X_2 = \{x : \max\limits_{i \in Q_2}[a_{i2}, x] = b_2\}, & \\ \qquad \cdots\cdots & x \in X. \\ x \in X_m = \{x : \max\limits_{i \in Q_m}[a_{im}, x] = b_m\}, & \end{array} \qquad (1)$$

*Работа поддержана грантом РФФИ № 06-01-00276.

Здесь X – выпуклое замкнутое ограниченное множество в $\mathbb{R}^n$, Q_i – некоторые множества из конечного числа элементов, $a_{ij} \in \mathbb{R}^n, b_i \in \mathbb{R}$. Квадратные скобки означают скалярное произведение. Данная задача называется *дискретной задачей квазилинейной алгебры.*

Заменяя каждое j-е ограничение на ограничение вида

$$x \in \overline{X}_j = \{x : \max_{u \in \overline{Q}_j} [a_{uj}, x] = b_j\},$$

где $\overline{Q}_j$ – компакт мощности континуум, получаем *непрерывную задачу квазилинейной алгебры.*

Замечание 1. Каждое уравнение можно представить в виде двух неравенств. Одно из неравенств (содержащее «$\leqslant$») определит выпуклый многогранник, второе – невыпуклое в общем случае множество. К системе уравнений можно добавить произвольную целевую функцию для решения методами из [6].

3. Общая схема метода

Нам понадобятся следующие определения.

Определение 1. Множество $Z \subseteq \mathbb{R}^n$ называется *конечно разрешимым*, если существует алгоритм линейного или выпуклого программирования, позволяющий найти его элемент за конечное время или установить его пустоту.

В частности, многогранные множества с рациональными коэффициентами конечно разрешимы.

Определение 2. Множество $Z \subseteq \mathbb{R}^n$ называется *ε-разрешимым*, если существует алгоритм линейного или выпуклого программирования, позволяющий найти его элемент с точностью до ε по евклидовой норме за конечное время или установить его пустоту.

Выпуклые ограниченные множества являются конечно разрешимыми, если известно минимальное содержащее их линейное подпространство, что зависит от способа задания множества. В противном случае они ε-разрешимы. В частности, если внутренность непуста, то выпуклое множество конечно разрешимо.

Определение 3. Множество $\overline{Z} \supseteq Z$ называется *релаксацией множества Z*, если оно выпукло и конечно разрешимо.

Алгоритм ветвей и границ для решения задачи (1) состоит из пяти составляющих, каждая из которых может быть реализована многими способами в зависимости от специфики задачи:

1) разбиение;
2) релаксация;
3) отсечение;
4) выбор перспективных областей;
5) критерий останова.

Алгоритм работает следующим образом.

Метод ветвей и границ для квазилинейной алгебры

Задается начальное разбиение

$$D_0 = \{X_1 \cap X, X_2 \cap X, \ldots, X_m \cap X\}.$$

{ **k-я итерация:** } пусть

$$D_k = \{X_1^k, X_2^k, \ldots, X_{n_k}^k\}.$$

Рассматриваются множества V_i точек, удовлетворяющих ограничениям

$$\begin{aligned} x \in V_1 &= \{x : \max_{i \in R_1 \subset Q_1} [a_{i1}, x] = b_1\},\\ x \in V_2 &= \{x : \max_{i \in R_2 \subset Q_2} [a_{i2}, x] = b_2\},\\ &\ldots\ldots\\ x \in V_s &= \{x : \max_{i \in R_m \subset Q_s} [a_{is}, x] = b_s\},\\ x \in X_s^k &\subset X, \quad s = 1, 2, \ldots, n_k. \end{aligned} \tag{2}$$

Предполагаем, что для каждого индекса i включение $R_i \subset Q_i$ строгое. Удобно выбрать $R_i = \{i\}$, т. е. фиксировать активный индекс. Тем не менее это не единственный возможный способ. Можно делить Q_i произвольно на два конечных множества индексов с количеством элементов, отличающимся не более чем на единицу. Построение множеств X_i^k зависит от X. Если X – параллелепипед, проще всего разбить его вертикальными и горизонтальными плоскостями, проходящими через центр. Разбиение допустимого выпуклого множества в методах ветвей и границ – стандартная задача (см. [3]).

Вычисляются релаксации множеств V_i для каждого i и отбрасываются пустые релаксации. Вычисляются точки из V_i – элементы релаксаций. Если хотя бы один элемент релаксации является решением исходной задачи, то ОСТАНОВ. Выбирается наиболее перспективное в некотором смысле множество $X_r^k \in D_k$ и разбивается как $X_r^k = X_{r1}^k \cup X_{r2}^k \cup \ldots \cup X_{rs}^k, X_{ri}^k \neq X_r^k$ для всех i, $X_{ri}^k \neq X_{rj}^k$ для всех i, j, $i \neq j$. Множества X_{ri}^k включаются в D_k, а X_r^k удаляется из него. Здесь $s \geqslant 2$ – количество новых элементов разбиения.

Для вычисления релаксаций можно использовать теорему Балаша [5].

Теорема 1 (Балаш, 1979). *Ограничение $\alpha x \leqslant \beta$ является суррогатом системы*

$$\vee_{i \in D} \; (A^i x \leqslant b^i), \quad x \geqslant 0,$$

тогда и только тогда, когда существуют $\Psi^i \geqslant 0$, $i \in D^*$, такие, что

$$\begin{aligned} \alpha &\leqslant \Psi^i A^i, \qquad i \in D^*, \\ \beta &\geqslant \Psi^i b^i, \qquad i \in D^*, \end{aligned}$$

где A^i – матрица размерности $m \times n$; Ψ^i – вектор $1 \times m_i$; $b \in \mathbb{R}^m$; D^* – подмножество индексов $i \in D$, для которого соответствующие ограничения с индексами i допустимы.

Суррогат системы означает, что соответствующее полупространство включает в себя решение системы (многогранное множество). Эквивалентно, это есть множество, задаваемое неравенством-следствием по отношению к *дизъюнктивной* линейной системе (содержащей связки ИЛИ).

Пересечение суррогатов является *релаксацией* для решения задачи. Алгоритм, основанный на данном подходе, впервые рассмотрен для произвольной дизъюнктивной задачи в [7].

Любые релаксации можно строить адаптивно, добавляя функции под максимумы и добиваясь более сильного отсечения.

Замечание 2. Можно вычислять выпуклую оболочку множеств V_i по формулам Ириа-Уррути и Лемарешаля [7, 8]. Они имеют следующий вид для множества вида

$$\vee_{j \in J} \; g_j(x) \leqslant 0,$$

определенного дизъюнкциями:

$$x = \sum_{j \in J} u_j,$$

$$\sum_{j \in J} \lambda_j = 1, \quad 0 \leqslant \lambda_j \leqslant 1, \quad j \in J,$$

$$\lambda_j g_j(u_j/\lambda_j) \leqslant 0, \quad j \in J.$$

Здесь g_j – произвольные выпуклые функции, что позволяет решать задачу (5) с выпуклыми функциями вместо линейных. Для линейных g_j можно сократить на λ_j, и следовательно, получаем конечно разрешимое множество. При таком подходе получаем алгоритм, близкий к описанному в [10] с той разницей, что мы имеем ограничения-равенства и можем по-другому организовать перебор.

Отсечение тривиально. Если релаксация данного множества пуста, то оно отбрасывается, поскольку не содержит решения. При этом следует проверять непустоту для каждой релаксации элементов D_k. Построение эффективного критерия останова также не вызывает трудностей. Пусть все множества в D_k конечно разрешимы. Тогда, проверив непустоту каждого из них, либо вычислим решение задачи (5), либо обнаружим пустоту допустимого множества. Если некоторое множество выпукло, оно совпадает с релаксацией.

Теорема 2. *Предположим, что на некоторой итерации все множества вида* (2) *совпадают со своими релаксациями. Пусть также любое текущее множество индексов R_i разбивается на подмножества, не совпадающие с R_i и в объединении дающие R_i. Тогда за конечное время будет найдено решение исходной задачи, если оно существует, или окажется, что допустимое множество пусто.*

Доказательство. Поскольку элементы разбиения совпадают с релаксациями, после полного перебора получим решение задачи, если оно существует. По построению решение не может быть исключено, так как отбрасываются только пустые релаксации и в силу выбора множеств R_i. Нахождение решения за конечное время следует из конечной разрешимости релаксаций. Пустота множества будет обнаружена за конечное время ввиду наличия всех элементов дизъюнкций.

Для совпадения множеств с релаксациями на некоторой итерации достаточно, чтобы множества X_i^k были выпуклыми многогранниками. Действительно, тогда множества в (2) выпуклы и конечно разрешимы начиная с некоторой итерации, поскольку множества R_i будут состоять из одного элемента.

Фиксируя последовательно разные индексы для построения R_i так, чтобы на наиболее низком уровне образующегося дерева оставались всевозможные комбинации индексов, получим, что условия данной теоремы выполняются.

Если множество Q_i выпукло и имеет мощность континуум (непрерывный случай), суть метода не меняется. В таком случае Q_i разбивается на два множества с тем, чтобы их меры стремились к нулю при последовательном разбиении. Аппроксимируя элементы разбиения описанными многогранниками, получаем на некотором этапе совокупность e-разрешимых множеств и приближенное решение исходной задачи.

Наиболее сложной частью является выбор перспективных областей для разбиения. Сходимость алгоритма за конечное число шагов от нее не зависит, но число итераций для нахождения решения задачи может быть очень большим при неудачном выборе. Эффективность способа выбора множества Y должна быть подтверждена многочисленными экспериментами. Предлагаются два метода оценки перспективности элемента разбиения.

1. *По максимальной невязке ограничений в релаксации.* Следует выбирать релаксацию с наибольшей по модулю разностью между правой и левой частями. Для произвольного многогранника $P = \{x \in \mathbb{R}^n \colon [c_i, x] \leqslant d,\ i \in I\}$ максимальная невязка ϕ вычисляется следующим образом:

$$\phi = \max_{i \in I}(d - [c_i, x]).$$

В дальнейшем выбирается релаксация с наибольшей максимальной невязкой.

2. *По мере релаксации.* Следует оценить меру с помощью приближенного вычисления интегралов и выбрать релаксацию наибольшей меры. Оценка перспективности релаксации V_i тогда равна $\int_{x \in V_i} dx$.

Сравнение метода ветвей и границ с модификациями алгоритма из [5] может быть проведено с помощью большого числа экспериментов.

Данный метод дизъюнктивного программирования может применяться для приближенного вычисления квазидифференциалов (см. [1, 2, 9]).

Будем искать аппроксимацию квазидифференциала некоторой функции f в точке x в виде $[V; W]$, где $V = \operatorname{co}\{v_1, v_2, \dots, v_m\}$, $W = \operatorname{co}\{w_1, w_2, \dots, w_r\}$ – выпуклые многогранники. Пусть мы имеем конечное множество направлений s_i, $i \in I$, где I – конечное множество индексов. Тогда задача оптимальной аппроксимации квазидифференциала по заданным направлениям имеет следующий вид:

$$\min_{V, W} \max_{i \in I} \Big| f'(x, s_i) - \min_{v \in V}[s_i, v] - \max_{w \in W}[s_i, w] \Big|. \tag{3}$$

Здесь $f'(x, s_i)$ – производная функции f в точке x по направлению s_i. Очевидно, минимум и максимум линейных функций на V и W достигаются в вершинах. Поэтому достаточно применить дихотомию по значению целевой функции и записать совокупность условий вида

$$\Big| f'(x, s_i) - \min_k [s, v_k] - \max_j [s, w_j] \Big| \leqslant t, \tag{4}$$

где t – меняющийся параметр. Задача (4) сводится тривиальным образом к общей задаче дизъюнктивного программирования. Ограничения-неравенства сводятся к равенствам введением дополнительной переменной, условие ее неотрицательности можно считать одним из ограничений X.

Остановимся подробнее на задаче непрерывной квазилинейной алгебры, которая не рассматривалась в [9,10]. Пусть она решается на допустимом множестве X, которое можно подразбивать на все более измельчающиеся выпуклые множества, например n-мерные кубы:

$$x \in \overline{X}_j = \{x : \max_{u \in \overline{Q}_j}[a_{uj}, x] = b_j\}, \quad j \in J,$$
$$x \in X.$$

Пусть множества $\overline{Q}_j$ выпуклы, ограниченны и телесны. Тогда существует алгоритм их разбиения, причем диаметры элементов разбиения стремятся к нулю, а объединение равно исходному множеству.

Тогда имеем *алгоритм двойного ветвления*. Во-первых, производится разбиение допустимого множества; во-вторых, разбиение множеств $\overline{Q}_j$, определяющих функции максимума.

В результате получаются дерево, в котором каждому узлу соответствует некоторое подмножество X, и некоторые подмножества $\overline{Q}_j$. Предположим, что все данные подмножества, если они не пусты (пустые отбрасываются), выпуклы и их диаметры стремятся к нулю при подразбиении.

Замечание 3. При небольшой размерности пространства проще использовать аппроксимации множеств $\overline{Q}_j$ многогранниками, переходя к последовательности дискретных квазилинейных задач. С ростом размерности, однако, число вершин таких многогранников быстро растет, ввиду чего предпочтительнее двойное ветвление.

Рассмотрим текущий узел дерева. Ему соответствует система уравнений и неравенств следующего вида:

$$\begin{aligned}
x \in X_1^p &= \{x : \max_{i \in U_1}[a_{u1}, x] = b_1\},\\
x \in X_2^p &= \{x : \max_{i \in U_2}[a_{u2}, x] = b_2\},\\
&\ldots\ldots\\
x \in X_m^p &= \{x : \max_{i \in U_m}[a_{um}, x] = b_m\},\\
x &\in \widehat{X}^p.
\end{aligned} \tag{5}$$

Здесь p – уровень узла дерева, U_j – элементы подразбиения.

Если множество X разбивается на многогранники с небольшим количеством вершин, получаем удобный критерий отсечения. Если какая-либо функция в уравнении строго положительна (проверяется вогнутой минимизацией, достаточен перебор вершин) или отрицательна (проверяется методами выпуклого программирования), то весь узел дерева отбрасывается, поскольку нет решений, соответствующих данным подмножествам.

Поскольку для достаточно больших уровней дерева диаметры подмножеств узла уменьшаются и близки к нулю, возможно, будет достаточно проверять значения функций максимума в одной точке. Лучше применить в окрестности решения алгоритм недифференцируемой оптимизации, поскольку такие алгоритмы эффективны для кусочно-линейных функций в непосредственной близости решения.

Если множества U_i, по которым вычисляется максимум, выпуклые и задаются гладкими функциями, то можно применить условия Куна–Таккера. В результате получается система задач дополнительности, причем ввиду подразбиения X ее можно решать стандартными алгоритмами, широко распространенными в настоящее время.

Целесообразно разбивать X и U_i на объединения измельчающихся симплексов. При этом для удобства можно переходить к обобщенным сферическим координатам.

Обозначим

$$S = \mathrm{co}_{i=1}^{n+1} v_i, \quad v_i \in \mathbb{R}^n,$$

симплекс в $\mathbb{R}^n$.

Если

$$S = S_1 \cup S_2 \cup \ldots \cup S_m,$$

где $S_i = \mathrm{co}\,\{v_1^i, v_2^i, \ldots, v_n^i\}$ и $v_j^i \in \mathbb{R}_+^n$ для всех $i = \overline{1, m}$ и всех $j = \overline{1, n}$, будем говорить, что $P = \{S_1, S_2, \ldots, S_m\}$ есть разбиение S. Если также $S_i \cap S_j = \varnothing$ при $i \neq j$, будем говорить, что P есть правильное разбиение S. Тогда, если мы имеем

$$\begin{gathered}
S_1 = S_{11} \cup S_{12} \cup \ldots \cup S_{1m},\\
S_2 = S_{21} \cup S_{22} \cup \ldots \cup S_{2m},\\
\ldots\ldots\\
S_m = S_{m1} \cup S_{m2} \cup \ldots \cup S_{mm},
\end{gathered}$$

будем говорить, что разбиение $\{S_{11}, S_{12}, \ldots, S_{mm}\}$ есть подразбиение P.

Можно рассматривать различные разбиения для сходимости. Простейшим и хорошо известным является разбиение по середине отрезка. Пусть

$$V = \mathrm{co}\,\{v_1, v_2, \ldots, v_n\}$$

есть симплекс с вершинами v_i. Положим

$$\|v_k - v_s\| = \max_{i,j} \|v_i - v_j\|.$$

Тогда разбиение следующее:

$$V = V_1 \cup V_2,$$

где

$$V_1 = \mathrm{co}\,(v_1, v_2, \ldots, v_{k-1}, (v_k + v_s)/2, v_{k+1}, \ldots, v_n);$$
$$V_2 = \mathrm{co}\,(v_1, v_2, \ldots, v_{s-1}, (v_k + v_s)/2, v_{s+1}, \ldots, v_n).$$

Тогда разбиение в алгоритме проводится согласно вышеуказанным правилам, причем множества изначально приводятся к виду объединения симплексов. Если данное представление неудобно, можно использовать и другие виды разбиений.

Личным вкладом автора является решение задачи квазилинейной алгебры с ограничениями-равенствами. Это важный в приложениях частный случай общей задачи дизъюнктивного программирования (которое включает в себя целочисленное программирование). При этом впервые предложен алгоритм решения задачи непрерывной квазилинейной алгебры. Алгоритм не совпадает с методами из [7, 10], так как учитывается специфика дизъюнктивной задачи, при этом могут использоваться известные способы построения релаксации дизъюнктивной системы и ее выпуклых оболочек. Результаты применены в квазидифференциальном исчислении для негладкого метода Ньютона [2].

Следует отметить, что рассматриваемая нами задача важна в негладком анализе и приложениях. Такие задачи возникают при минимизации негладких функций (см. [1,2]). Функции максимума весьма часто встречаются в математической экономике.

4. Заключение

В данной работе предлагаются схемы решения важной задачи – системы уравнений сублинейной алгебры. Такие проблемы возникают в негладком анализе, в различных технических приложениях. Построенные алгоритмы позволяют находить глобальные и локальные оптимумы сложных недифференцируемых функций, решать задачи, возникающие в динамических системах. Особенно большое применение алгоритмы могут найти в моделях с кусочно-дифференцируемыми, в частности квазидифференцируемыми, функциями [1,7]. Предполагается исследовать полиномиальность алгоритмов решения задачи, которая, вообще говоря, проще общей задачи дизъюнктивного линейного программирования.

Автор благодарит В. Ф. Демьянова за поддержку данного исследования.

Литература

1. Андрамонов М. Ю. Методы глобальной минимизации для некоторых классов обобщенно выпуклых функций. Казань: Казан. матем. о-во, 2001.
2. Демьянов В. Ф. Теорема о неподвижной точке в негладком анализе и ее применение. СПб.: Изд-во СПбГУ, 1996.
3. Гороховик В. В., Зорько О. И. Полиэдральная квазидифференцируемость вещественнозначных функций // Докл. АН Беларуси. 1992. Т. 36, № 5. С. 393–397.
4. Еремин И. И. Сигма-кусочные функции и задачи дизъюнктивного программирования // Докл. РАН. 1998. Т. 358, № 5. С. 538–540.
5. Balas E. Disjunctive programming // Annals of Discrete Mathematics. 1979. №. 5. P. 3–51.
6. Balas E., Tama J., Tind J. Sequential convexification in reverse convex and disjunctive programming // Mathematical Programming. 1989. Vol. 44, № 3. P. 337–350.
7. Beaumont F. Algorithm for disjunctive programming problems // European J. Oper. Res. 1990.
8. Demyanov V. F., Rubinov A. M. Constructive Non-Smooth Analysis. Frankfurt: Peter Lang, 1995.
9. Grossmann I., Lee S. A review of mixed-integer nonlinear programmming // Computers and Chemical Engineering. 2000.
10. Grossmann I., Lee S. New algorithm for nonlinear generalized disjunctive programming // Ibid. Vol. 24, № 9. P. 2125–2142.
11. Handbook on Global Optimization / Eds. R. Horst, P. Pardalos. Dordrecht: Kluwer Academic Publishers, 1996.